

```

rdf:_2                rdfs:range        rdfs:Resource .
...

```

Offensichtlich sind die axiomatischen RDFS-Tripel in mehrere Gruppen aufgeteilt. Die erste Gruppe enthält nur Tripel mit `rdfs:domain` als Prädikat. Deklativer „Zweck“ eines solchen Tripels `p rdfs:domain c` ist es, dass man die URI `p` mit einem Klassenbezeichner `c` verknüpft. Dadurch wird eine Zugehörigkeit zu dieser Klasse (vermittelt durch `rdf:type`) dann jeder URI `s` „aufgezwungen“, die als Subjekt gemeinsam mit dem Prädikat `p` in einem Tripel `s p o` auftritt.

So sagt beispielsweise das fünfte Tripel in dieser Auflistung nichts anderes, als dass, wann immer ein Tripel `c rdfs:subClassOf d` auftritt (welches bekanntlich aussagt, dass `c` Unterklasse von `d` ist), sofort folgt, dass `c` auch eine Klasse ist, ausgedrückt durch das Tripel `c rdf:type rdfs:Class`.

Analog dazu bewirken die in der zweiten Gruppe zusammengefassten Tripel mit Prädikat `rdfs:range` eine Typisierung von Tripel-Objekten.

Bezüglich offener Listen wird durch die axiomatischen Tripel einerseits die Klasse der Enthaltenseins-Beziehungen als Unterklasse aller Propertytypen spezifiziert, andererseits `rdfs:Container` als Überklasse aller bekannten Arten offener Listen festgelegt.

Sodann wird die `rdfs:isDefinedBy`-Property deklariert als Spezialisierung der `rdfs:seeAlso`-Property. XML-Literal wird als Datentyp und Unterklasse aller Literalwerte kenntlich gemacht, außerdem wird die Klasse aller Datentypen als Klasse von Klassen identifiziert.

Schließlich werden den Listenelement-Propertytypen die ihnen gebührenden Eigenschaften zugewiesen.

Basierend auf dem Begriff der RDFS-Interpretation definieren wir nun analog zu den vorhergehenden beiden Fällen, dass ein Graph G_2 aus einem Graphen G_1 *RDFS-folgt*, wenn jede RDFS-Interpretation, die Modell von G_1 ist, auch Modell von G_2 ist.

4.2.4 Interpretation von Datentypen

Wir wissen bereits, dass es in RDFS nur einen einzigen vordefinierten Datentyp gibt, nämlich `rdf:XMLLiteral`, dessen semantische Eigenschaften vollständig durch die im vorigen Abschnitt eingeführten RDFS-Interpretationen abgedeckt sind. Gleichwohl können extern definierte Datentypen in RDF(S) verwendet werden.

Wie in Abschnitt 3.3.1 bereits besprochen, lässt sich ein Datentyp d beschreiben als bestehend aus einem *Wertebereich* Val_d , einem *lexikalischen Bereich* Lex_d und einer Funktion $Lex2Val_d$, welche jedem Element des lexikalischen